Public Comments on NIST Draft Special Publication 800-132

NIST received the following public comments on the draft Special Publication 800-132, “Recommendation for Password-Based Key Derivation Part I: Storage Applications (June 2010)”.

The comments are ordered based on the dates they were received. Most comments were received in e-mail format. The line-breakers are deleted to make them more readable. The e-mail headers are removed to protect commenter’s privacy. For the same reason, e-mail address and telephone number are also removed if signature is included. The following is a list of the commenter name and the page number of the comments.

Terence Spies

Do you know if NIST looked at the Halting KDF protocols for this document? The idea here is to randomize (through user input) the number of hash passes, and store a confirming value. This prevents active dictionary attacks since the attacker doesn't know how many rounds of hashing were used in the derivation. The paper on this actually establishes that you get additional bits of key strength through the use of this process.

More details here:

http://ai.stanford.edu/~xb//security07/index.html

Terence

Henrick Hellström

The security considerations that led to PKCS#5 are obsolete.

The main difference between a PBKDF and any KDF, is that the former features an Iteration Count, that is meant to increase the complexity of the operation and thereby increase the security strength of the derived key by a few "virtual" bits of security. This was an important feature back in the days when it was easy to memorize passwords with, say 40 bits of entropy, and adding the equivalent of 16 bits of complexity was a significant security improvement.

However:

1. Using the iteration count to increase the strength of a password with 40 bits of entropy to a key with 128 bits of security is obviously infeasible, since it would require 2^{88} iterations.

2. There is no real need for a PBKDF that allows you to add 16 bits of complexity to a Pass Phrase with 112 bits of entropy. There is no reliable function for measuring the exact entropy of a pass phrase with such a small margin of error. In order to be reliably confident you have a pass phrase with 112 bits of entropy, you need a pass phrase that is likely to have at least twice that amount of entropy. Since the user has to be encouraged to select very strong pass phrases anyway, adding a few bits of complexity will not add any significant amount of security.

Conclusion:
NIST should focus on methods for measuring the entropy of pass phrases. If strong pass phrases are used, there is no apparent reason to use a PBKDF with an iteration count over any other approved KDF

Kok-Wah LEE

Dear Sir/Madam,

The due date to comment for SP 800-132 (SP = Special Publication) on the "DRAFT Recommendation for Password-Based Key Derivation - Part 1: Storage Applications" is by 28Jul2010.

Please refer to page 11 of the draft SP 800-132 for sentence "Easily accessed personal information, such as the user's name, phone number, and date of birth, should not be included in a password", where I have my opinions over here to comment.

Yes, hereby I would like to inform some new password management techniques [T1-T3] proposed by me in the breakthroughs [BT1-BT2] to have solved some critical problems in key management, that may have affected the human habits to create password the secret key.

[BT1] Memorizable key size
Using 2D key (Two-Dimensional Key) [L1], one can achieve the range of memorizable key sizes from 128 bits to 256 bits, and beyond for persons with extraordinary memory power.

[BT2] Number of slave keys per master key Using multihash key [L1] and YinYang-1000 memory card [L2], a user can have a memorizable master key to control up to 1000 unique slave keys. As from survey, one can normally remember four to five secret keys, so it is about 4000 to 5000 slave keys for quite abundant number of online and offline accounts.

[L1] Memorizable Public-Key Cryptography (MePKC) & Its Applications 2D key: Section 2.4 Multihash key: Section 7.2 Multi-factor multimedia token key: Section 5.3
http://www.archive.org/details/MemorizablePublic-keyCryptographymepkcItsApplications

[L2] Implementation of 2D key
http://www.xpreeli.com/2D_Key.htm

{T1] First new technique of password management is to use 2D key in [BT1] to create memorizable password the key over 128 bits.

[T2] Second new technique of password management is to use multihash key in [BT2] to generate multiple slave keys from a master key, where this master key can be a 2D key. This second technique is similar to the password-based key derivation in this draft SP 800-132, but specifically selected algorithmic steps have been arranged to create slave keys from a master key instead.

[T3] Third new technique of password management is to optionally embed the user's ID (Identity) into a 2D key possibly used together with multihash key, like names, passport number, phone number, email, birthday, birth place, etc., preferably to have uniquely or selectively sieved out for the identification of password owner. This technique is alike the technique of public key certificate (aka digital certificate) to bind together the user's identity and public key. Then, the advantage to recognize the account owner is there in case of ID theft (i.e. identity theft) to have hacked the online and offline accounts.

Consequently, please consider revising the sentence "Easily accessed personal information, such as the user's name, phone number, and date of birth, should not be included in a password." at page 11 of the draft SP 800-132. In short, personal identity can be partial part(s) of a password, but not as a whole.

Here, I hope the people reading the final copy of this SP 800-132 are assisted by my added informative comments.

Thanks and Bye.

Regards
Kok-Wah LEE @ Xpree Li
Information Engineer
Xpreeli Enterprise

Walt Hubis

I would like to see greater clarification regarding handling of the additional variables required to recover the DPK that is briefly discussed in the very last paragraph of this document. Specifically:
The additional variables needed to recover a DPK using the mechanisms specified in [3-5] (e.g., initialization vectors) may be stored on the hard drive or another storage device. For example, can the initialization information (salt, key length, etc.) be stored in the clear in a non-secure location? Or, is some level of protection required for this information?

Regards,
Walt Hubis
Software Architect
Engenio Storage Group

Claudia Popa

Classification: UNCLASSIFIED

Good afternoon,

Please find attached few comments for the draft NIST SP 800-132.

<<SP 800-132_Comments_Claudia Popa.doc>>

Best regards,
Claudia Popa

1. Page 1, Introduction

"This Recommendation specifies a family of Password-Based Key Derivation Functions (PBKDFs) for deriving cryptographic keys from passwords for the protection of electronically-stored data."

Section 5.4 specifies that Master Key can be used as Data Protection Key (DPK) and as a key used to protect DPKs generated through other methods.

The sentence above could include this info:

"This Recommendation specifies a family of Password-Based Key Derivation Functions (PBKDFs) for deriving cryptographic keys from passwords for the protection of electronically-stored data or for the protection of data protection keys."

2. Page 1, Introduction

"This Recommendation specifies a family of Password-Based Key Derivation Functions (PBKDFs) for deriving cryptographic keys from passwords for the protection of electronically-stored data."

I try to understand how the requirements of this Special Publication apply to the CMVP testing.

A cryptographic module validated by the CMVP could implement the requirements of this SP, generate a key from a password, and output this key. This key could be used for the protection of electronically-stored data, but it could be used for other reasons. If the key is output from the boundary of the cryptographic module, how this key is used is outside the scope of the CMVP.

Is it expected that the CMVP will not allow the generation of a key from a password if this key is exported/output from the boundary of the cryptographic module?

3. Page 5, General Discussion

*"If the MK is used to protect DPKs, the protection **shall** use an **approved** authenticated encryption mode, such as defined in [3-5], or an **approved** key protection method."*

It is not clear what is considered NIST **approved** key protection method. The document has to clarify what are considered NIST approved key protection methods.

4. Page 5, Password-Based Key Derivation Functions

This section identifies minimum length requirements for C (Counter) and S (Salt). There is no requirement for the length of the password.

IEEE 802.11i, in Annex H, H.4 Suggested pass-phrase to PSK mapping, includes a method of deriving keys from a passphrase. This standard recommends a length of minimum 20 characters for the pass-phrase.

Appendix A of this Special Publication, A.1 User-Selected Passwords, states that passwords shorter than 10 characters are usually weak, but does no actually include any requirement about the minimum length of a password used for key derivation.

Is it not important to specify a requirement for the minimum length for a password?

5. Page 8, Option 1

I am not sure what this paragraph is saying. Is this information relevant for this Special Publication?

"For options 1a and 1b, if only encryption is applied to the plaintext data, and the data size is large, in order to detect an incorrect entry of the password, the plaintext data might include some redundancy that can be checked easily without decrypting the whole data on the storage medium."

6. Page 7, Using the Derived Master Key to Protect Data

*"In both options, a DPK is used to protect electronically-stored data, and the correctness of the MK **shall** be verified."*

I expect that after this Special Publication is approved the CMVP will update the FIPS 140-2 Implementation Guidance and the CMVP will allow the generation of keys from a password if the requirements of this SP are met. A Computer Security Testing laboratory will perform the testing for the CMVP.

How will a tester verify this requirement?

7. Page 9, Option 2

"In the second option, randomly generated DPKs are protected (e.g., encrypted) in two different ways."

I don't believe that **(e.g., encrypted)** is required in this sentence.

The paragraph after this sentence actually defines how the DPKs have to be protected.

8. Page 10

"The use of an ***approved*** *authenticated encryption mode or key protection method allows the detection of an incorrect MK or incorrect derived keying material and, by extension, an incorrect password, thus avoiding the lengthy process of decrypting the protected data using an incorrect DPK. "*

The same comment as #3 above. The document has to clarify what are considered NIST approved key protection methods.

9. A.3, Protection of DPK

The same comment as #3 above. The document has to clarify what are considered NIST approved key protection methods.

T. Wayne Nugwin

Hello,

Please see the attached Comment Grid.

Thank you.

Wayne Nugwin
Bureau of Labor Statistics

Comments for <NIST SP 800-132 Draft, Recommendation for Password-Based Key Derivation, Part 1: Storage Applications>
Agency: <Bureau of Labor Statistics, OTSP-DNIA>

Section	Comments	Recommendations	Resolution/Action to be taken
Section 5.4, page 9, Figure 3	Option 2b in the in the diagram shows an input "Decrypted DPK" to the "Decryption" box. The "Option 2" text description below Figure 3 does not elaborate on this input.	Clarify on the input "Decrypted DPK" and its role/function in deriving the keying material to subsequently protect the DPK (or Data Protection Key). If there is no relevant function, then remove "Decrypted DPK" from diagram to avoid confusion with "DPK" as ultimately generated to protect electronically-stored data.	

Andrey Jivsov

Dear NIST:

I would like to provide the feedback on the time line of the adoption of this method, presently undefined, and possible interoperability issues.
There is also a suggestion for achieving 112 bit minimum security at the end of this note.

Certain products at Symantec adopt different derivation function, which is Iterated and Salted String to Key (S2K), defined in RFC 4880. It is an iterative function based on any FIPS-approved Hash function. SHA-1 is typically used, but the transition to SHA-3 when it becomes a standard is allowed by data structures (I will note that SHA-1 is Acceptable by SP 800-131 draft for this non-signature application).

In my opinion, the functionality of S2K is equivalent to that of PBKDF, in particular:
* both use Salt to thwart dictionary attacks
* both are iterative to slow down the derivation with a dynamic iteration count (iteration count is stored in the public data structures)
* both methods have similar weakness: they don't include iteration count into the hashing process, enabling additional dictionary attacks in protocols and applications in which Salts are predictable

The change in password derivation function is critical to product usability because the change affects long-term data structures stored on media, such as private keys. We will need to clear the following milestones to accomplish full transition to PBKDF in PGP products:

* OpenPGP IETF standard doesn't yet define PBKDF support. It may take at least a year to have the standard published. Typically IETF requires at least two interoperable implementations to advance a draft to an RFC (per RFC 2026).
* The ability to read new data format in earlier versions of applications in generally desirable. We would like to wait for reasonable customer adoption of new PGP products with PBKDF read-only support first.
* Only after this we can start delivering the products that are fully compliant with PBKDF.

Given these factors outside of our control, it will be unfeasible to be fully compliant with this method in less than 2 years. There is additional benefit to delay this work until SHA3 is defined.

An aggressive timeline would substantially limit the product choice available to the Government, favoring the products that happen to presently use PBKDF.

The last issue in this comment is about observed perception that overestimates benefits of PBKDF. I observed that the SP 800-132 may lead some to believe that PBKDF operation in itself can enhance the strength of the password to the level that the passwords can become equivalent with 128-bit keys. In reality, PBKDF only adds log2(C) bits of entropy to the password strength (16 bits for minimum recommended C), where C is the Iteration Count. Once the delay attributed to C is within user-noticeable level, doubling it for each additional bit of entropy is unpractical. A related issue is the estimation of the password entropy:
because the strength of the password will be the most important factor in determining the strength of the derived master key, it appears challenging how compliance with SP 800-131 will be established, if there is a plan to allow keys derived from passwords.

Thank you.
Andrey Jivsov
PGP, Now a part of Symantec Corporation

Darren Lasko

To whom it may concern,

The Trusted Computing Group’s Storage Working Group has the following comment on the draft version SP800-132:

The minimum iteration count (C) specified in the document is too high for storage devices that implement the PRF in firmware on low-powered microcontrollers, leading to an unacceptable user experience of long authentication times.

Sincerely,
The members of the Trusted Computing Group Storage Working Group

---

Vijay Bharadwaj

Thank you for publishing this draft for comment. We agree with the objective of providing a standard key derivation method based on user-memorable secrets, as this is an important scenario in practice.

Here are our comments on the draft:

1. This draft of SP 800-132 is subtitled “Part 1: Storage Applications”. However the general discussion describes a generic capability for password-based key derivation instead of the more targeted use of password-based key derivation for

2. The Information Assurance Directorate at the National Security Agency has published draft Common Criteria protection profiles security functional requirements for full disk encryption and encryption for USB flash drives. However, those two documents and the draft SP 800-132 have slightly different requirements for deriving the data protection key.
   - The draft protection profiles permit both passwords and passphrases. We believe that SP 800-132 should also be worded to include passphrases as well as passwords. There is reason to believe that passphrases are often more convenient to users than passwords – they are typically longer than passwords with the same entropy but they are also easier to remember. The PBKDF2 algorithm works equally well for passphrases, so it seems unnecessary to exclude them.

   - There are differences in the recommended password lengths.

3. In general, we believe that this document would be improved by including a more detailed security analysis involving security aims and how the design satisfies them. In particular:
   - What is the assessed security strength of this key derivation mechanism, as a function of the PRF, salt length and iteration count? This would be useful for users trying to understand how secure a password-based scheme might be versus a different scheme. It will also become important if NIST were to revise the FIPS 140 Implementation Guidance to lift the prohibition on using password-based key derivation in FIPS mode.

   - As an example of the above, the recommended salt length should be tied to the target security strength instead of being arbitrarily specified as 128 bits.

   - Further, the choice of the underlying hash for the HMAC should also be based on the target security strength.

   - The iteration counts recommended in the draft are somewhat arbitrary and may be too high for many applications. As a point of reference, Windows currently uses an iteration count on the order of tens of thousands, and this runs in about 100ms on our minimum recommended hardware. This is already in the range of user-noticeable delays and can be annoying in some applications. It would be better to supplement the specific numbers

- It would be useful to provide pointers to guidance related to passwords, so that users could see how to deploy software that uses PBKDFs. This could be in the form of references to other SPs that discuss password selection policies, password strength, etc.

Thanks,

Vijay Bharadwaj, Mike Grimm and Mike Lai
Microsoft Corporation